BÁO CÁO THỰC HÀNH LAB 3

LẬP TRÌNH HƯỚNG ĐỐI TƯỢNG

Contents

[1. Đề bài 1](#_Toc150985470)

[2. Yêu cầu bài toán 1](#_Toc150985471)

[3. Mã nguồn 2](#_Toc150985472)

[3.1 Working with method overloading 2](#_Toc150985473)

[3.2 Passing parameter 3](#_Toc150985474)

[3.3 Classifier Member and Instance Member 4](#_Toc150985475)

[3.4 Write new methods to implement the following functions 6](#_Toc150985476)

[3.5 Implement the **Store** class 8](#_Toc150985477)

[3.6 Re-organize your projects 10](#_Toc150985478)

[Hình 1 Overload Cart.java adđigitalVideoDisc 3](#_Toc150985487)

[Hình 2 Kết quả và code của swap 4](#_Toc150985488)

[Hình 3 Code của thêm ID vào DigitalDisc 5](#_Toc150985489)

[Hình 4 Code của thêm ID vào DigitalDisc 6](#_Toc150985490)

[Hình 5 Hàm search 7](#_Toc150985491)

[Hình 6 Kết quả của hàm search 8](#_Toc150985492)

[Hình 7 Tạo Store 9](#_Toc150985493)

[Hình 8Tạo Store 9](#_Toc150985494)

[Hình 9 Test Store 10](#_Toc150985495)

[Hình 10 Code sau khi refactoring 10](#_Toc150985496)

# Đề bài

Thiết kế hệ thống mới cho hệ thống AIMS. (Hiện tại chỉ có 1 phương tiện: DVD)

# Yêu cầu bài toán

Đối với người mua hàng:

* Duyệt danh sách các DVD có sẵn trong cửa hàng
* Tìm kiếm DVD theo: tiêu đề, danh mục và giá cả
* Xem thông tin chi tiết của 1 DVD
* Thêm DVD vào giỏ hàng
* Xem giỏ hàng
* Sắp xếp DVD trong giỏ hàng theo tiêu đề hoặc chi phí
* Cập nhật số lượng DVD trong giỏ hàng
* Đặt hàng

Đối với quản lý cửa hàng

* Đăng nhập, kiểm tra quyền
* Xem danh sách các đơn hàng đang chờ xử lý
* Xem chi tiết  đơn hàng và xử lý nốt(đồng ý/từ chối đặt hàng)
* Thêm(xóa) DVD mới vào(ra) danh sách sản phẩm

# Mã nguồn

## 3.1 Working with method overloading

**Open the class Cart.java:** you will overload the method **addDigitalVideoDisc** you created last time.

+ The current method has one input parameter of class **DigitalVideoDisc**

+ You will create a new method that has the same name but with different type of parameter.

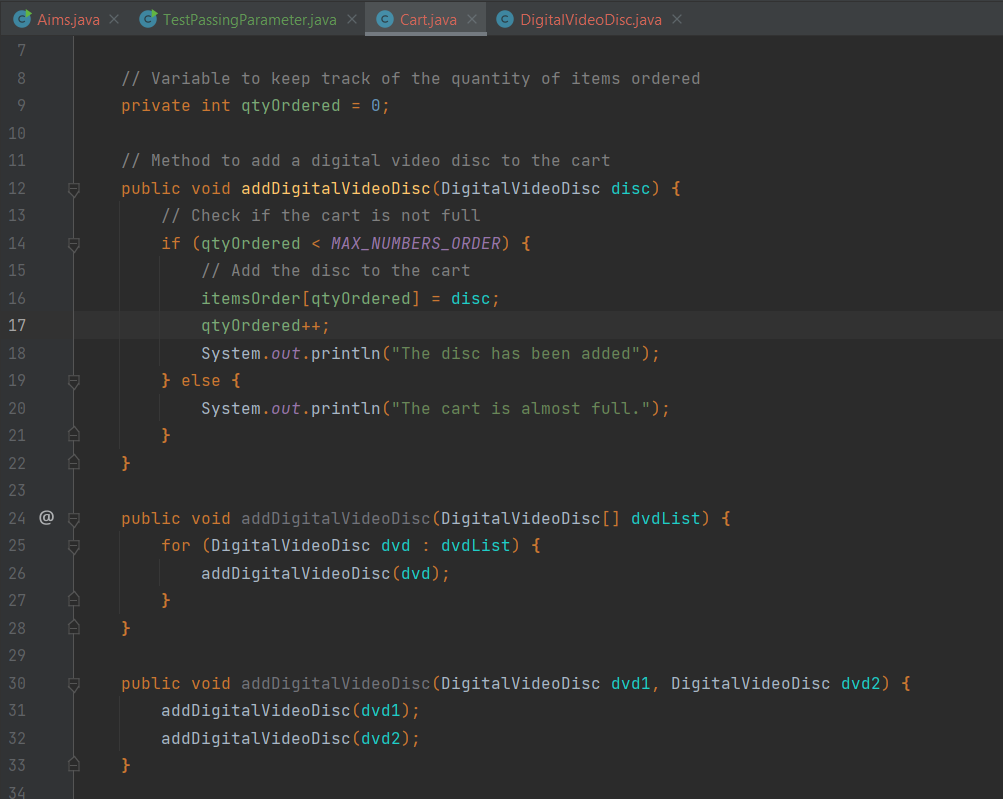
**addDigitalVideoDisc(DigitalVideoDisc [] dvdList)**

This method will add a list of DVDs to the current cart.

+ Try to add a method **addDigitalVideoDisc** which allows to pass an arbitrary number of arguments for dvd.

Compare to an array parameter. What do you prefer in this case?

* Tôi thích dùng truyền mảng hơn. Vì khi gọi hàm, có thể truyền cả mảng thay vì gõ từng dvd vào một



Hình 1 Overload Cart.java adđigitalVideoDisc

## Passing parameter

***Is JAVA a Pass by Value or a Pass by Reference programming language?***

- Java is a "pass-by-value" programming language

First of all, we recall what is meant by **pass by value** or **pass by reference**.

* Pass by value: The method parameter values are **copied** to another variable and then the copied object is passed to the method. That's why it's called pass by value
* Pass by reference: An alias or reference to the actual parameter is passed to the method. That's why it's called pass by reference.

Now, you will practice with the **DigitalVideoDisc** class to test how JAVA passes parameters. For this exercise, you will need to temporarily add a setter for the attribute title of the DigitalVideoDisc class.

Create a new class named **TestPassingParameter** in the current project

To test whether a programming language is passing by value or passing by reference, we usually use the **swap** method. This method aims to swap an object to another object.

* After the call of **swap(jungleDVD, cinderellaDVD)** why does the title of these two objects still remain?
* Vì trong hàm swap chỉ đổi giá trị o1 và o2 cho nhau, không gây ảnh hưởng đến nội dung, và không thay đổi object mà 2 biến ở hàm main hướng đến.
* After the call of **changeTitle(jungleDVD, cinderellaDVD.getTitle())** why is the title of the JungleDVD changed?
* Vì hàm làm thay đổi nội dung của object mà dvd trỏ đến. Đồng thời việc gán dvd = new (oldTitle) không làm thay đổi jungleDVD trỏ đến cái dvd mới này.

Kết quả: A screenshot of a computer program

Description automatically generated

Hình 2 Kết quả và code của swap

## 3.3 Classifier Member and Instance Member

**Open the DigitalVideoDisc class:**

* You should note that this class only has instance variables: **title**, **category**, **director**, **length**, **cost**.

- Now, we know that each DVD has a unique id assigned by the system. One simple way to manage all the ids is to give them out to new DVDs as consecutively incremented values. In order to do this, we must keep track of the number of DVDs created.

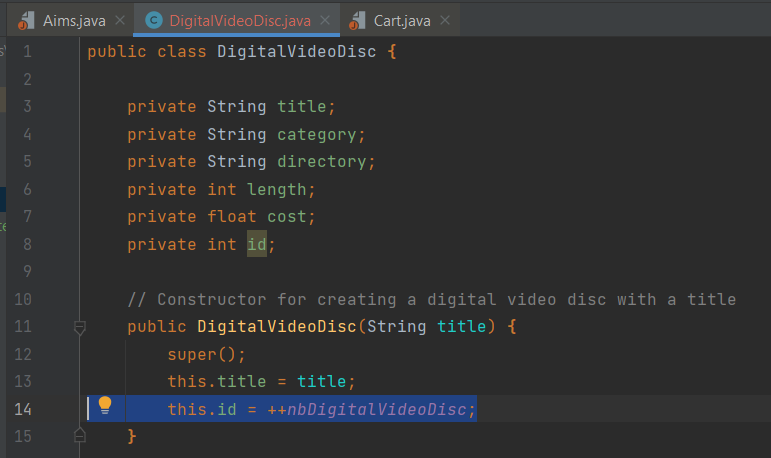
- Create a class attribute named "**nbDigitalVideoDiscs**" in the class **DigitalVideoDisc**

private static int nbDigitalVideoDiscs = 0;

- Create an instance attribute named "**id**" in the class **DigitalVideoDisc**

- Each time an instance of the **DigitalVideoDisc** class is created, the **nbDigitalVideoDiscs** should be updated. Therefore, you should update the value for this class variable inside the constructor method and assign the appropriate value for the **id**.

Tạo thêm biến **id** vào trong class **DigitalVideoDisc:**



Hình 3 Code của thêm ID vào DigitalDisc

A screenshot of a computer program

Description automatically generated

Hình 4 Code của thêm ID vào DigitalDisc

3.4 Write new methods to implement the following functions  
- Create a new method to print the list of ordered items of a cart, the price of each item, and the total price. Format the outline as below:

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*CART\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Ordered Items:

1. DVD - [Title] - [category] - [Director] - [Length]: [Price] $

2. DVD - [Title] - ...

Total cost: [total cost]

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

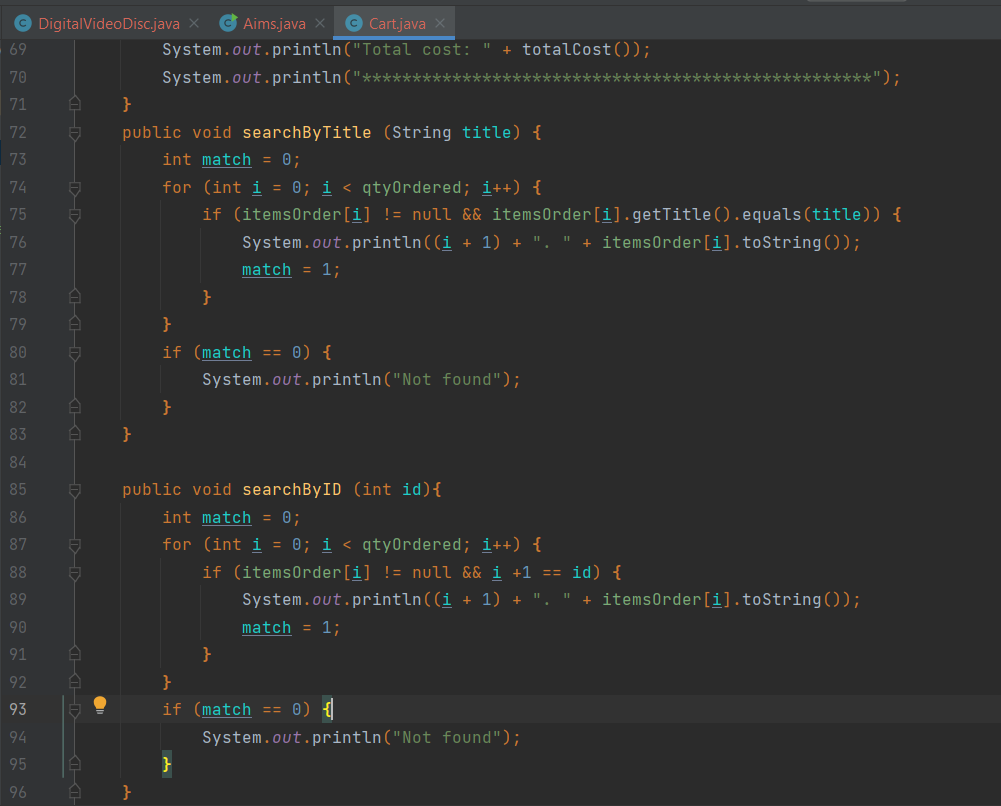
**Suggestion**: Write a **toString()** method for the **DigitalVideoDisc** class. What should be the return type of this method?

- Hàm trả về String

- Search for DVDs in the cart by ID and display the search results. Make sure to notify the user if no match is found.

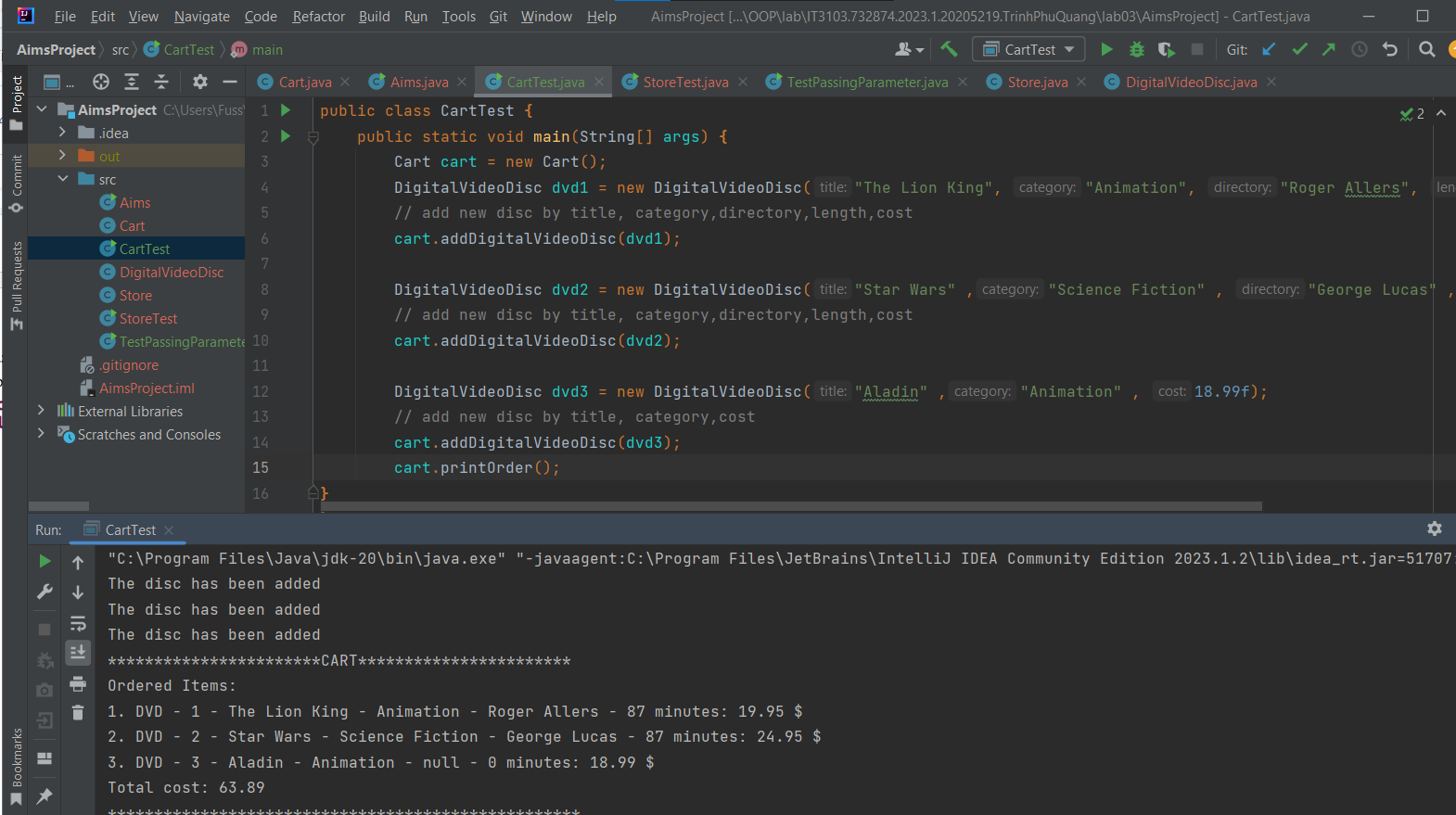
- Search for DVDs in the cart by title and print the results. Make sure to notify the user if no match is found. ***Refer to problem statement in Lab02 for the matching rule.* Suggestion**: write a **boolean isMatch(String title)** method in the **DigitalVideoDisc** which finds out if the corresponding disk is a match given the title.

- In the **CartTest** class, write codes to test all methods you have written in this exercise. You should create sample DVDs and carts, like in this code snippet:



Hình 5 Hàm search

Kết quả:



Hình 6 Kết quả của hàm search

## 3.5 Implement the **Store** class

- Create a **Store** class, which contains one attribute **itemsInStore[]** – an array of DVDs available in the store.

- To add and remove DVDs from the store, implement two methods called **addDVD** and **removeDVD**

- Test these two methods in **StoreTest** class.

Store class:

A computer screen shot of text

Description automatically generated

Hình 7 Tạo Store

A screen shot of a computer program

Description automatically generated

Hình 8Tạo Store

StoreTest class:

A screenshot of a computer program

Description automatically generated

Hình 9 Test Store

## 3.6 Re-organize your projects

A screenshot of a computer

Description automatically generated

Hình 10 Code sau khi refactoring